THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

GRAPHICS PROPERTIES HOLDINGS, INC., Plaintiff, v. ASUS COMPUTER INTERNATIONAL, INC., Defendants.	Civil Action No. 12-cv-210-LPS Jury Trial Demanded
GRAPHICS PROPERTIES HOLDINGS, INC., Plaintiff, v. TOSHIBA AMERICA INFORMATION SYSTEMS, INC., and TOSHIBA CORPORATION, Defendants.	Civil Action No. 12-cv-213-LPS Jury Trial Demanded
GRAPHICS PROPERTIES HOLDINGS, INC., Plaintiff, v. VIZIO, INC., Defendant.	Civil Action No. 12-cv-214-LPS Jury Trial Demanded

GRAPHICS PROPERTIES HOLDINGS, INC., Plaintiff, v. GOOGLE, INC. Defendant.	Civil Action No. 12-cv-1394-LPS Jury Trial Demanded
GRAPHICS PROPERTIES HOLDINGS, INC., Plaintiff, v. HEWLETT-PACKARD COMPANY, Defendant.	Civil Action No. 12-cv-1395-LPS Jury Trial Demanded
GRAPHICS PROPERTIES HOLDINGS, INC., Plaintiff, v. LENOVO GROUP LTD., et al., Defendant.	Civil Action No. 12-cv-1397-LPS Jury Trial Demanded
GRAPHICS PROPERTIES HOLDINGS, INC., Plaintiff, v. ASUS COMPUTER INTERNATIONAL, INC., et al., Defendants.	Civil Action No. 13-cv-864-LPS Jury Trial Demanded

EXPERT DECLARATION OF DR. WILLIAM H. MANGIONE-SMITH IN SUPPORT OF GRAPHICS PROPERTIES HOLDINGS, INC.'S CLAIM CONSTRUCTION BRIEF REGARDING U.S. PATENT NO. 5,717,881

1. My name is Dr. William Henry Mangione-Smith. Counsel Graphic Properties Holdings Inc. ("GPH"), the Plaintiff, has retained me as an expert consultant in the above-identified cases. I have been asked by GPH to offer this expert declaration on the issue of claim construction with respect to U.S. Patent 5,717,881 ("the '881 Patent"). I have been asked to review the'881 Patent and related materials and to provide my expert opinion based on that review. More specifically, I have been asked to perform an analysis of multiple terms within the asserted claim and consider the proper construction for those terms from the point of view of a person of ordinary skill in the art.

2. In reaching the opinions and conclusions set forth in this declaration, I have relied upon my education, experience and training, my review of the patent and its prosecution history, as well as my review of the evidence produced in this matter.

3. I reserve the right to supplement this declaration if further information becomes available or if I am asked to consider additional information. Furthermore, I reserve the right to consider and comment on any additional statements and testimony of Defendants' experts in this matter. I also may rely on demonstrative exhibits at trial to explain my testimony and opinions.

4. I am the sole proprietor and employee of Phase Two, LLC and I am being compensated at $550/hour for my work related to this declaration. My compensation is in no way dependent on the outcome of this matter or the contents of this declaration, the testimony or opinions that I provide. I expect to be compensated at the same rate for any time spent testifying by deposition or at trial.

I. EXPERT QUALIFICATIONS

5. My curriculum vitae and testimony list are included in Exhibit A. To summarize my qualifications, I hold three academic degrees in the field of Computer Engineering: A Bachelor's of Science in Engineering, a Master's of Science in Engineering, and a Doctorate of Philosophy degree. All of my degrees were earned at the University of Michigan in Ann Arbor. I have been involved in the design of computer systems, microprocessors, firmware and software applicable to a broad range of systems for more than twenty years.

6. My technical background covers most aspects of computer system design, including low level circuitry, computer architecture, computer networking, graphics, application software, and system software (e.g., operating systems and compilers). Early on in my career I had a particular focus on high performance computing. For the past 20 years my attention has been directed towards consumer and embedded devices such as cell phones, laptop and desktop computers, and various control systems. I am a member of the Institute of Electrical and Electronics Engineers and the Association for Computing Machinery, which are the two most significant professional organizations in my field. I have been employed as a design engineer, research engineer, professor and technical expert. Over my professional career I have been an active inventor with 58 issued U.S. patents, 166 published and pending U.S. patent applications and additional unpublished U.S. patent applications.

7. From 1984 until 1991 I attended the University of Michigan in Ann Arbor, Michigan. I was awarded the degrees Bachelors of Engineering, Masters of Engineering, and Doctorate of Philosophy in the area of computer engineering. My doctoral research focused on high performance computing systems including computer architecture, applications and operating system software, and compiler technology. One of my responsibilities included

teaching senior Bachelors of Engineering and graduate students who were about to enter the profession.

8. After graduating from the University of Michigan I was employed by Motorola in Schaumburg, Illinois. While at Motorola, I was part of a team designing and manufacturing the first commercial battery-powered product capable of delivering Internet email over a wireless (i.e., radio frequency) link and one of the first personal digital assistants. I also served as the lead architect on the second-generation of this device. Part of my responsibilities at Motorola involved the specification, design, and testing of system control Application-Specific Integrated Circuits ("ASICs"). I conducted the initial research and advanced design that resulted in the Motorola M*Core embedded microprocessor. M*Core was designed to provide the high performance of desktop microprocessors with the low power of contemporaneous embedded processors. In particular, M*Core was most often used in battery powered systems where low power consumption was a vital design goal. While at Motorola I was the sole inventor on a single U.S. patent.

9. From 1995 until 2005 I was employed by the University of California at Los Angeles ("UCLA") as a professor of Electrical Engineering. I was the director of the laboratory for Compiler and Architecture Research in Embedded Systems ("CARES") and served as the field chair for Embedded Computing Systems. The CARES research team focused on research, engineering and design challenges in the context of battery-powered and multi-media mobile computing devices. My primary responsibility, in addition to classroom teaching, involved directing the research and training of Masters of Engineering and Doctoral candidates. I was a tenured member of the faculty, and had responsibilities for teaching as well as scholarly research. While at UCLA I was a named inventor on one U.S. patent and two pending patent

applications. My colleagues at UCLA were some of the leading scientists and engineers in the world with a long list of innovations from computer network security devices to the nicotine patch. The graduate student researchers in my laboratory came from a diverse set of backgrounds, all with bachelor's degrees in computer engineering, electrical engineering or computer science, many with multiple years of experience working as professional engineers in areas such as software development, computer system design and ASIC circuit design.

10. I was employed at Intellectual Ventures in Bellevue, Washington from 2005 until 2009. My responsibilities at Intellectual Ventures included business development, technology assessment, market forecasting, university outreach, collaborative inventing, intellectual property licensing support, and intellectual property asset pricing. My colleagues and co-inventors at Intellectual Ventures included the former lead intellectual property strategist at Intel, Intel's former lead IP council, Microsoft's former chief software architect, the founder of Microsoft research, the designer of the Mach operating system, the architect of the U.S. Defense Department's Strategic Defense Initiative, the founder of Thinking Machines which was one of the seminal parallel processing computer systems, and Bill Gates. I had responsibility for hiring and managing over 15 staff members including multiple Ph.Ds. with degrees in electrical engineering and decades of experience in product design and engineering.

11. Since the beginning of 2009, I have been working as an independent consultant in a wide range of technology areas related to innovation and intellectual property licensing.

II. RELEVANT LEGAL STANDARD

12. I am neither a lawyer nor a patent agent. However, certain legal principles that relate to my opinions have been explained to me.

A. Standard for Claim Construction

13. I understand that a patent includes a specification that describes the patented invention. The written specification includes a description of the preferred embodiment of the invention, drawings and claims that define the scope of the patent. When interpreting the claims, I understand that the ordinary meaning of the language within the claims should be followed unless the specification or prosecution history provides reason for a different interpretation. I further understand that the claims may not be interpreted to require particular features of the preferred embodiment of the invention unless the claims expressly recite those features. I understand that, in this case, the parties have submitted their proposed claim constructions to the Court and in this report I am giving my opinion with regards to each set of proposals.

14. It is my understanding that the terms used within a claim should be construed to have the meaning that a person of ordinary skill in the art would employ when reading the claims. However, as mentioned above, such a reading must be made within the context of the entire intrinsic record. Thus, for example, if the patentee decides to clearly define a term in an unconventional manner that definition should be honored when reading the patent. Furthermore the use of various claim terms within the patent specification should be given significant weight when interpreting their meaning with extrinsic evidence generally carrying lesser weight.

15. It is my understanding that 35 USC § 112 (6) is the controlling law when a patent claims a means for or step for structure. I have been informed that the mere use of the phrase “means for” or “step for” does not necessarily invoke the constraints of 35 USC § 112 (6), however such language does indicate that 35 USC § 112 (6) is likely applicable. As I understand it a claim using “means for” language under 35 USC § 112 (6) must be evaluated using a two-

step process. First, one must clearly identify the function that is implemented by the claimed means. Second, one must identify the structure within the patent body that implements that claimed function. I also have been informed by counsel that literal infringement of a claim with means-plus-function terms requires that the accused product perform the identical function as the claimed function and contain the same or equivalent structure as the identified corresponding structure, and the equivalent structure must have been available at the time of the issuance of the claim.

16. I have been informed that a preamble is considered a claim limitation when it is necessary to breathe life into a claim.

B. Standard for Definiteness

17. I understand that a patent claim meets the definiteness requirement if one of ordinary skill in the art would understand the scope of the patent claim. I also understand that a claim is indefinite if the claim is "insolubly ambiguous, and no narrowing [claim] construction can be adopted." Further, I understand that claim elements written in a means-plus-function manner can be indefinite if there is an insufficient disclosure of the structure corresponding to the claimed function.

III. LEVEL OF ORDINARY SKILL IN THE ART

18. When interpreting the claim terms and assessing infringement, I have attempted to interpret the '881 Patent from the perspective of a person of ordinary skill in the relevant art. Based upon my review of the '881 patent, the patent's file history and my experience as an expert I have reached the following opinions regarding the context of the claimed inventions.

19. It is my opinion that the field of art is hardwired microprocessors. In my opinion, a person of ordinary skill in the art at the time of the invention (at least June 11, 1990)

of the ‘881 Patent would normally have the knowledge acquired by a person having a Bachelor’s degree in Electrical Engineering, Computer Engineering, or a related degree. Such a person would also have 3 to 5 years of work experience or a Master’s degree in Electrical Engineering, Computer Engineering, or related degree working on the design of hardwired microprocessors or related fields. I believe that a person of ordinary skill can obtain the requisite knowledge from on-the-job experience, training, education, or a combination of these sources. Based on this definition of the level of ordinary skill in the art, I consider myself at least one of ordinary skill in the art. Further, I was one of ordinary skill in the art at the time of the ‘881 Patent.

IV. OVERVIEW OF THE ‘881 PATENT

20. The Cray-1 Supercomputer, first installed at Los Alamos National Labs in 1976, was a very successful product for Cray Research and was recognized for providing extremely high performance on both vector and scalar operations. However, it was generally recognized that the Cray-1 had a particular limitation when it came to executing scalar instructions. Each instruction on the Cray-1 could be either one parcel in length or two parcels. When the Cray-1 issued a two parcel instruction a single clock cycle would occur prior to the issuance of the subsequent instruction. This delay occurred regardless of whether the data and function units required by that second instruction were available or not. The ‘881 Patent claims an invention that addressed this limitation and allowed the possibility for issuing a mixture of two and one parcel instructions where each clock cycle corresponded to an instruction being issued. This capability greatly improved the peak performance of the Cray computer architecture on scalar processing.

V. CLAIM CONSTRUCTION

21. I have reviewed the claim construction developed by GPH, as well as that proposed by the Defendants. To the extent that GPH’s proposed construction is materially

different from that of Defendants, I have compared and analyzed each set of constructions. I list each of these below. In addition, I summarize the basis for my conclusions that one construction is preferable over the other.

A. Hardwired Supercomputer

Claim Language	GPH's Construction	Defendants' Construction
"hardwired supercomputer"	This term does not require construction. However, to the extent the Court disagrees that this term does not need construction, it should be construed as: "hardwired high performance computer." *See* "hardwired."	A hardwired computer that is among the fastest or most powerful of those available at a given time.

22. GPH and Defendants have offered up separate constructions for the word "hardwired" and the term "hardwired supercomputer." The parties have suggested constructions for "hardwired supercomputer" that explicitly employ the word "hardwired". Thus, I will address the meaning of "hardwired" below, and at this point focus on the difference between proposed constructions for the larger term.

23. GPH suggests "hardwired supercomputer" need not be construed and I agree with them. In my opinion the word "supercomputer," which occurs in the preamble of claim 1 from the '881 Patent, does nothing to breathe life into the claim. The history of computer systems have consistently shown a migration of technologies from very high performance and specialized computers, such as supercomputers, down into mainframe computers, desktop computers, personal computers and eventually embedded devices. The migration of technologies discussed within the '881 Patent specification such as "branch prediction" and the ability to simultaneously decode multiple instructions from supercomputers into all forms of computing devices illustrates that the technologies are not solely related to supercomputers. At the time the '881 Patent was developed by Cray Research Incorporated the company was solely identified with the design, development and manufacture of

supercomputers. That being the context of development it is not surprising to me that the inventors thought of the technology as related to supercomputers. However the technology described and disclosed in the ‘881 Patent is now generally found embedded into microprocessors within individual integrated circuits.

24. Another perspective on this trend is the fact that the phrases “supercomputer on a chip” and “mainframe on a chip” return over 75 thousand web pages from a Google search including links to devices developed decades ago by companies such as Intel and IBM. The practical reason for this fact is that supercomputers have traditionally been defined by the ability to produce extremely high performance on scientific applications. Scientific applications are generally characterized as mathematically intensive, largely based on linear algebra to manipulate data in matrix form, and based upon floating-point data representations. When the inventors of the ‘881 Patent developed their claimed technology such capability generally required many hundreds of integrated circuits, tens of circuit boards, and multiple physical storage racks to house all of this equipment. The same capability can be found today in a range of high-performance individual integrated circuits. Thus, in many cases, modern microprocessors are supercomputers.

25. GPH’s alternative construction of “hardwired supercomputer” is supported by the intrinsic evidence, which provides that “most prior art supercomputers have attempted to increase both the performance and the functional capabilities of the individual computer processors in such supercomputers.” *See* Plaintiff Graphics Properties Holdings, Inc.’s Opening Claim Construction Brief (“GPH Brief”) at Exhibit 1, ‘881 Patent, at 2:1-4. This passage from the specification notes that supercomputers were designed to increase performance of modern computers. Thus, a construction of “supercomputer” as “high-performance computer” is both

accurate and logical. Moreover, a person of ordinary skill in the art understands "supercomputer" to mean high-performance computer. GPH's alternative construction is the most logical interpretation to those of ordinary skill.

26. In contrast, the Defendants suggest that the term "hardwired supercomputer" be interpreted to mean "A hardwired computer that is among the fastest or most powerful of those available at a given time." Defendants seem to be arguing for a temporal quality to the term, which would mean a Cray supercomputer suddenly becomes something else, not a supercomputer, simply by the passage of time. From my experience a person of ordinary skill in the art would not use the term in such a manner.

B. Instruction Fetch Means For Providing An Instruction Stream Of Two Parcel Items In Sequence, Wherein Each Two Parcel Item Has A Bit Length Of 2n

Claim Language	GPH's Construction	Respondent's Construction
instruction fetch means for providing an instruction stream of two parcel items in sequence, wherein each two parcel item has a bit length of 2n	This phrase should be construed pursuant to 35 U.S.C. § 112, ¶ 6. Function: Provide an instruction stream of two parcel items in sequence, wherein each two parcel item has a bit length of 2n. Structure: Program counter 126 coupled to instruction cache 110 over an address bus.	This term should be construed in accordance with 35 U.S.C. § 112, ¶6. Recited Function: Providing an instruction stream of two parcel items in sequence, wherein each two parcel item has a bit length of 2n Corresponding Structure: Instruction fetch port 112, instruction cache 110, and program counter 126, as described at 4:48-52, 6:19-7:43, Figs. 2, 5a, and 5b.

27. I believe that this claim term should be construed as GPH has proposed, not as Defendants have. The parties agree as to function, but not structure.

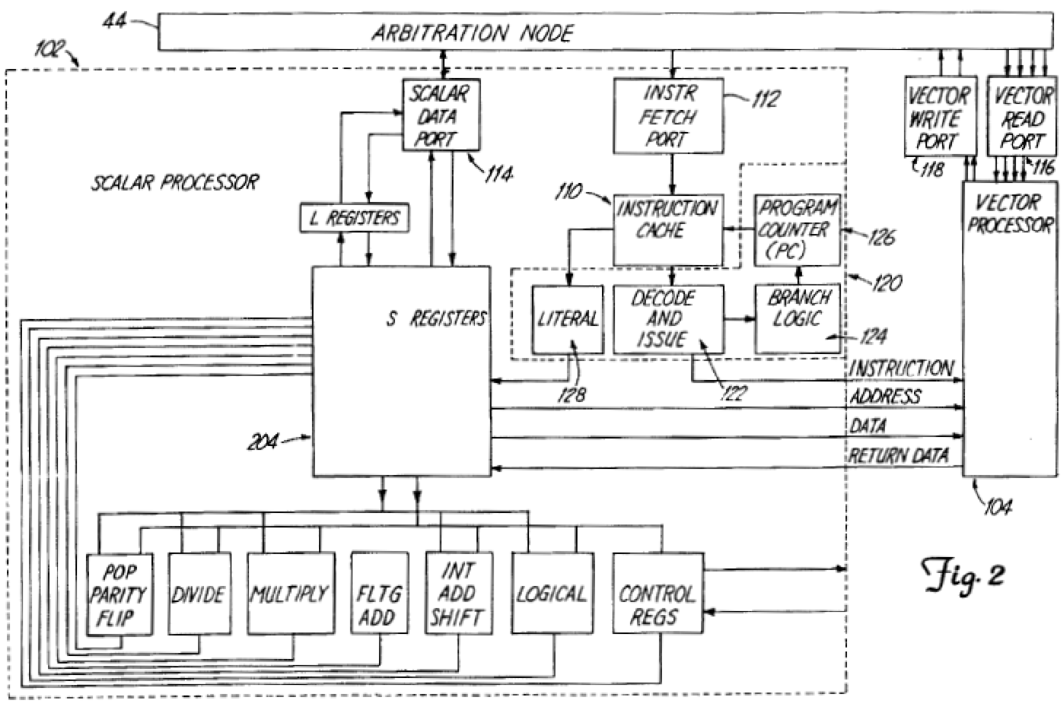
28. All of the parties agree that the instruction fetch means structure includes a program counter 126. This is supported by the specification. The '881 Patent specification recites "An instruction fetch means including a program counter 126 transfers instructions in sequence to a pipeline including register holding elements 201 and 202 which form a pipeline between the instruction cache and the instruction decode and issue mechanism 122." (GPH Brief

at Exhibit 1,‘881 Patent at 9:36-41). The ‘881 Patent appears to have clearly stated that the instruction fetch means includes a program counter. This element comports with my understanding of how computer systems work. While there exists a small number of mostly theoretical or research computers without program counters, I am not aware of any such systems that were actually manufactured, and Cray Research certainly never manufactured such a system. The purpose of a program counter is to identify the next instruction for execution and the first step in executing an instruction is fetching it. Thus a person of ordinary skill would understand that a program counter would be a component of an instruction fetch means, as the inventors state it is for the ‘881 Patent.

29. However, the Defendants’ construction is flawed for at least the reason that it includes an instruction fetch port that the inventors do not associate with the instruction fetch means. Moreover, Defendants’ inclusion of this element is inappropriate because this structure is unnecessary to carry out the claimed function of the instruction fetch means. The parties agree that the function of the instruction fetch means is “providing an instruction stream of two parcel items in sequence, wherein each two parcel item has a bit length of 2n.” The ‘881 Patent specification makes it plain that this instruction stream is provided to the instruction decode means, and not to the instruction cache through the fetch port 112.

30. In particular, the ‘881 Patent provides that “[a]n instruction fetch means including a program counter 126 transfers instructions in sequence to a pipeline including register holding elements 201 and 202 which form a pipeline between the instruction cache and the instruction decode and issue mechanism 122.” (GPH Brief at Exhibit 1, ‘881 Patent at 9:36-41). Fig. 2 of the ‘881 Patent (reproduced below) expressly shows that the instruction stream is provided by the output of instruction cache 110, and not by the fetch port 112, to the instruction

decode and issue mechanism 122. This is also consistent with the language of the claim itself, which requires the instruction decode means, rather than the instruction cache to be responsive to the instruction stream. (*See* GPH Brief at Exhibit 1, '881 Patent at claim 1 ("instruction decode means responsive to each two parcel item[.]"))



(*Id.* at Fig. 2.)

31. Those skilled in the art would recognize that coupling a program counter to an instruction cache, as illustrated in Fig. 2 of the '881 Patent, provides all of the necessary structure that is required for providing an instruction stream of two parcel items in sequence, wherein each two parcel item has a bit length of 2n. Indeed, the '881 Patent expressly provides that "the PC register 126 (FIG. 2) contains a 32-bit word address and is used to fetch the 64-bit words out of the instruction cache 110. Words are fetched from the instruction cache 110 at a rate of up to one per clock cycle as needed by the instruction execution unit 120." Thus, by

addressing the instruction cache, the PC causes the instruction cache to output the claimed instruction stream. Nothing more is needed.

32. Defendants' construction is flawed because it requires a very particular organization of instruction cache 110 that is unnecessary to the claimed function. There can be no serious dispute that instruction caches were well known in the art. Indeed, the '881 Patent recognized that "in a *typical* instruction cache, a single smaller block of instructions is loaded into a faster access cache hardware to decrease the access time." (GPH Brief at Exhibit 1, '881 Patent at 6:40-44.) In a preferred embodiment, the '881 discloses an internal organization of instruction cache 110 as a four-way associative buffer. *Id.* at 6:19-7:43, Figs. 5a and 5b.) However, as the inventors correctly recognized, the purpose of the internal organization of the cache "as a four-way associative buffer [was to] allow[] the instruction cache 110 of the preferred embodiment to minimize both instruction fetch times and cache thrashing." However, neither the organization of the cache as a four-way associative buffer nor minimizing fetch times and cache trashing have anything to do with the claimed instruction fetch means function. *Id.* at 6:49-53.

33. Consequently, it is my opinion that the construction proposed by GPH would serve to accurately construe the "instruction fetch means" term.

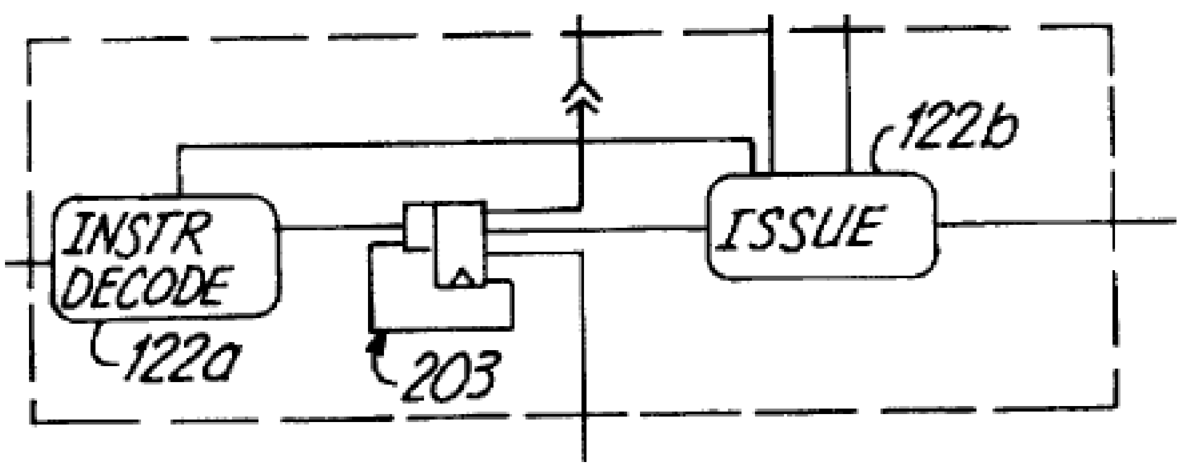
C. Instruction Decode Means Responsive To Each Two Parcel Item For Determining In One Clock Cycle Whether The Two Parcel Item Is A Single Two Parcel Instruction Having A Bit Length Of 2n Bits Or Two One Parcel Instructions, Each Having A Bit Length Of N Bits

Claim Language	GPH's Construction	Defendants' Construction
instruction decode means responsive to each two parcel item for determining in one clock cycle whether the two parcel item is a single two parcel instruction having a bit length of 2n bits or two one parcel instructions, each having a bit length of n bits	This phrase should be construed pursuant to 35 U.S.C. § 112, ¶ 6. Function: Responsive to each two parcel item, determine in one clock cycle whether the two-parcel item is a single two parcel instruction having a bit length of 2n bits, or two one-parcel instructions,	This term should be construed in accordance with 35 U.S.C. § 112, ¶6. Recited Function: Responsive to each two parcel item, determining in one clock cycle whether the two parcel item is a single two parcel instruction having a bit parcel instruction having a bit length of 2n bits or two one parcel instructions, each having a bit length of n bits

	each having a bit length of n bits. Structure: Hardwired decode mechanism 122a an embodiment of which is further described by 3:4-8, 4:59-62, 9:41-44, 27:43 - 61, 33:1-5, 33:44-46, 35:16-20, Appendices A-B, and Figs. 2 and 7C.	Corresponding Structure: The specification does not disclose a corresponding structure, therefore, this limitation is indefinite under 35 U.S.C. § 112. <u>Google:</u> Recited Function: Responsive to each two parcel item, determining in one clock cycle whether the two parcel item is a single two parcel instruction having a bit parcel instruction having a bit length of 2n bits or two one parcel instructions, each having a bit length of n bits. Corresponding Structure: This limitation is indefinite under 35 U.S.C. § 112. The patent identifies the following blocks as performing instruction decode, though these blocks cannot perform the recited function: Fig. 2; the decode units (122a, 129a, 129b) of decode and issue mechanism 122, depicted in Figs. 7C-7D and described at 4:59-5:3, 27:53-61, 33:1-5.

34. GPH has identified the instruction decode mechanism 122a as the corresponding structure. I agree with this position. There can be no serious disagreement regarding the fact that 122a is the claimed instruction decode means. There is no ambiguity that 122a corresponds to the instruction decode means – the specification makes it clear that it does. For example, the specification states “[s]imultaneously, the trap instruction goes into the issue mechanism through the decode means 122a.” ‘881 Patent at 33:44-45; *see also id.* at 33:2-3 (identifying 122a and 122b as “the instruction decode and issue means 122a and 122b[.]”) The patent further states “An instruction execution unit 120 in the scalar means 102 includes decode and issue means 122, branch logic means 124, a program counter (PC) register 126 and literal transfer means 128.” (GPH Brief at Exhibit 1, ‘881 Patent at 4:59-62). The patent also claims “An instruction fetch means including a program counter 126 transfers instructions in sequence to a pipeline including register holding elements 201 and 202 which form a pipeline between the instruction cache and the instruction decode and issue mechanism 122. The issue mechanism includes a decode portion 122a and an issue portion 122b.” (*Id.* at 9:36-42)

35. A portion of Fig. 7c is shown below. This figure shows the instruction being fed into block instruction decoder 122a from the left with the output of instruction decoding emerging from the right hand side and moving forward to the issue unit 122b. The bottom portion of Fig. 7c (not shown here) indicates that data produced by the instruction decoder 122a during clock cycle –S1 is later used by the issue mechanism 122b at clock cycle zero. Note that the bottom of Fig. 7c shows "-52 -51 0" when a person of ordinary skill in the art would understand the figure to be presenting what occurs in various pipeline stages, namely the zero stage, the stage prior to that ("-S1"), and the next previous stage ("-S2"). This representation of instruction decode capabilities is consistent with the understanding of a person of ordinary skill in the art, as illustrated by the two subsequent images created by very well-known computer architects illustrating the decode capabilities in essentially the same manner.



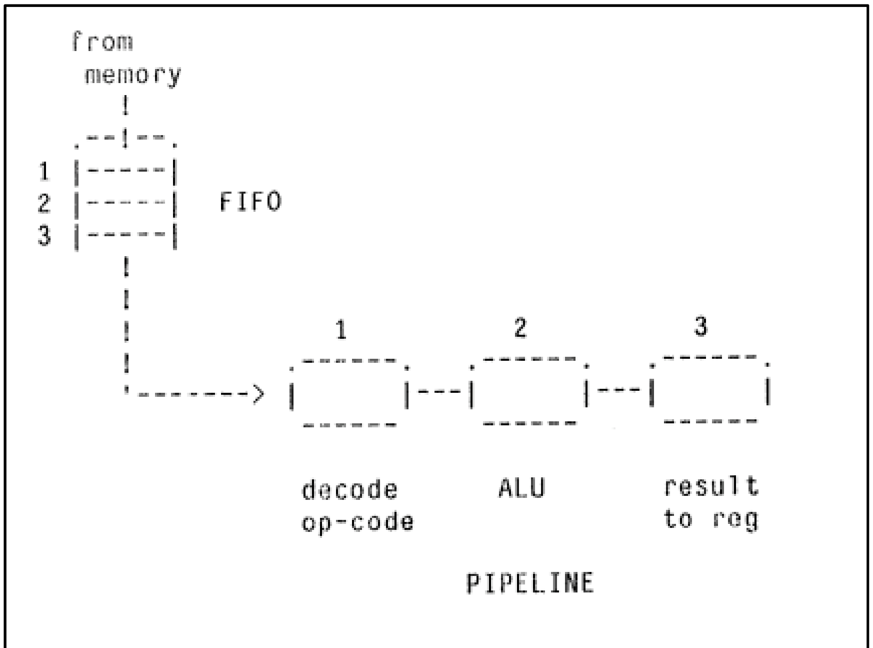
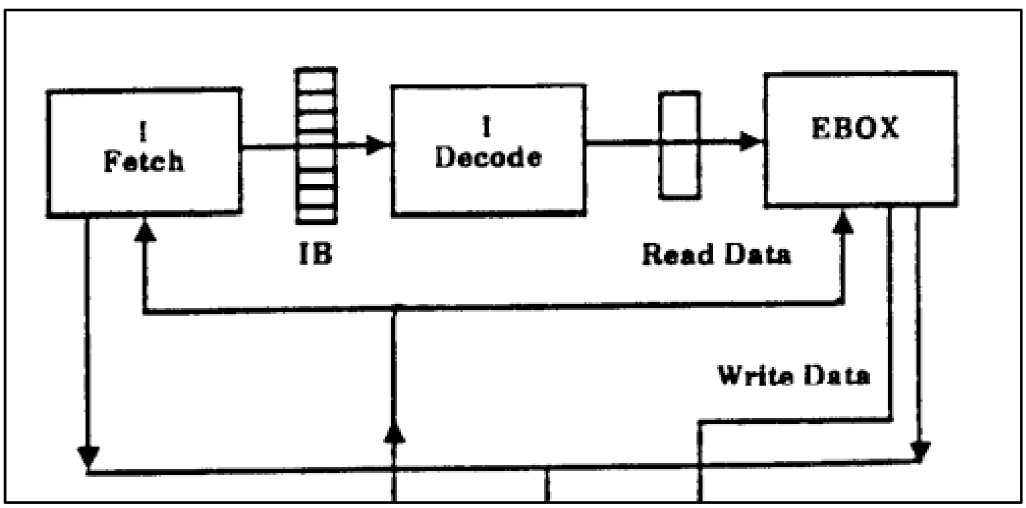


Figure 1

(*See* "Keeping Jump Instructions Out Of The Pipeline Of A Risc-Like Computer", Maurice V. Wilkes, ACM SIGARCH Computer Architecture News, Volume 11 Issue 5, December 1983, Pages 5-7 ACM New York, NY, USA (produced asGPH_DE_01696206 – GPH_DE_01696208) (attached hereto as Exhibit B).



(See "A Characterization of Processor Performance in the VAX-1 l/780", Joel S. Emer and Douglas W. Clark Proceedings of the 11th International Conference on Computer Architecture, May 1984 (produced as GPH_DE_01696209 – GPH_DE_01696218) (attached herein as Exhibit C).

36. The patent further discloses how the instruction decode mechanism 122a accomplishes the identified function: “The present improvement decodes all 64 bits at once, examines the instruction fields, determines whether the 64 bits are one or two instructions, determines which instruction goes where and whether it goes on the first or second issue cycle.” (GPH Brief at Exhibit 1, ‘881 Patent at 27:55-59) This passage comports with the decoder illustrated in Fig. 7c as well as the instruction formats illustrated in Fig. 4a-4c. Each of the disclosed instruction formats have a well-defined opcode field (labeled op in the sequence Fig. 4a-4c) allowing a simple decoder to be constructed that can implement the claimed function using design techniques that were well known to a person of ordinary skill in the art at the time the ‘881 Patent was invented.

37. Defendants’ proposal that 129a and 129b correspond to the *claimed* instruction decode means is misguided. While the ‘881 patent specification may have identified more than one instruction decode means, as shown below, the digital logic decoder circuit, 122a, is the *only* instruction decode means that is coupled to the instruction issue means as required by the plain language of claim 1. (*See Id.* at ‘881 Patent Claim 1) (requiring an “instruction issue means *responsive to the instruction decode means*”. - emphasis added.)

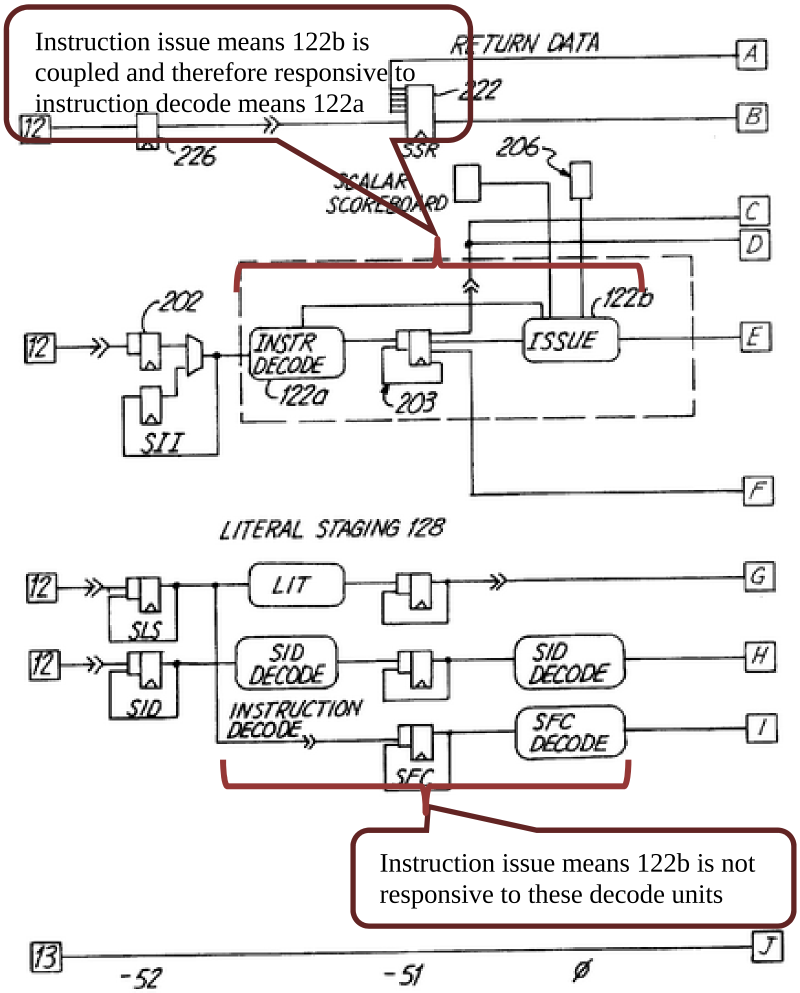


Fig. 7C

38. Defendants begin by claiming that the term is indefinite because suitable structure is not disclosed. I disagree and it is my opinion that the discussion above presents suitable structure for the claim term.

39. As a preliminary matter, Defendants' position that the '881 Patent does not disclose suitable structure is misplaced. The '881 Patent clearly constrained the instruction decode means structure to a hardwired instruction decode circuit. See Excerpts of '881 Patent File History, March 19, 1997 Amendment and Response to Office Action & Examiner Interview Summary, produced as GPH_DE_01511506 – GPH_DE_01511512 at 6 (attached hereto as

Exhibit D) ("When the control functions of a digital computer system are generated by hardware using hardware logic design techniques, the control unit is said to be 'hardwired.") In theory, if what Defendants' propose is true, then the '881 Patent would allow the function corresponding to the instruction decode means to be carried out by *any* structure, such as an analog circuit, a piece of software, or a mechanical device – but it does not.

40. To the contrary, the '881 Patent expressly requires the structure corresponding to the instruction decode means to be a hardwired instruction decoder circuit. (*See* GPH Brief at Exhibit 1 at, Fig. 7C (illustrating "INSTR DECODE" 122a interconnected with digital logic components such as flip flops and registers and synchronized with a digital clock); *see also Id.* at 3:37:41 ("FIGS. 7, 7A-7H and their organization in FIG. 7 is a block diagram which illustrates the primary components of the improved processor, the data and logic flow through the components in the relative time sequences in which the components are active."); *see also Id.* at Claim 1 ("[a] hardwired supercomputer data processing apparatus"); *see also Id.* at ("register holding elements 201 and 202 which form a pipeline between the instruction cache and the instruction decode and issue mechanism 122"); *see also* Id. '881 Patent at 9:42-44 ("[a] single stage holding means 203 couples the decode and issue mechanisms"); *see also* Joint Proposed Claim Constructions at p. 3 (the parties agree that the term hardwired, as it appears in claim 1 of the '881 patent means "[t]he control functions for a digital computer system are generated by hardware using hardware logic design techniques."); *see also* '881 Patent Microfiche Appendices A – B, produced as GPH_DE_01696219 – GPH_DE_01696550("the '881 Patent Microfiche"), GPH_DE_01696221 – 529 (disclosing the detailed instruction set architecture of the digital decoder circuit); *see also* Exhibit D, Excerpts of '881 Patent File History, Appeal Brief, produced as GPH_DE_01511465 – GPH_DE_01511501 ("The

specification of the present application dearly describes such a hardwired supercomputer data processing apparatus having a structure including a hardwired control unit to perform the control functions of the supercomputer with hardware logic design techniques. See Figures 1-2 and the corresponding text of the specification and detailed system overview from page 16, line 30 through page 21, line 27 and the supporting Figures 7A-7H.”).

41. Other than a hardwired instruction decoder circuit, the ‘881 Patent recognizes no other structure (e.g., analog circuits, piece of software, mechanical or chemical function generators) that would be capable of carrying out the claimed function, and that would otherwise be entirely permissible for carrying out the claimed function in a so-called “black box.”

42. Schematically, the ‘881 Patent depicts the hardwired instruction decoder circuit structure as a rectangular symbol because that is the standard way that is used in the art to draw a hardwired instruction decoder circuit. This symbolic representation of a hardwired instruction decoder circuit is widely used in undergraduate textbooks, academic publications, IEEE standards, and in the industry.

43. Now is a worthwhile time to consider how electrical engineers designed devices with digital logic at the time of invention for the ‘881 Patent. Typically a designer would have a toolbox of components from which to choose. During the technology tutorial, I discussed a number of common components well known to one of ordinary skill in the art. An n-bit AND gate receives n input values and produces an output of zero in all cases except the single case where all of the inputs have a value of one. Similarly, an n-bit OR gate receives n input values and produces an output of one in all cases except the single case where all of the inputs have a value of zero.

44. One additional component in the toolbox of every person of ordinary skill in the arts is the n-bit digital logic decoder. The two passages below from undergraduate textbooks published just before the '881 Patent was drafted clearly teach that a decoder has n-bits of input and activates (sets to the value of one) one and only one of 2^n output signals. Thus, if 4-bits of data are provided on the input there will be 16 output lines and exactly one of these lines will be asserted (take the value of one) at any time. These passages also teach that, schematically, a digital logic decoder is to be represented by a rectangular symbol that is equivalent to the rectangular symbol used to represent the digital logic decoder structure of the '881 Patent.

transmission gates are bidirectional, these multiplexers can be used as "demultiplexers," or decoders. We will discuss them next.

EXERCISE 8.17

Show how to make a 4-input multiplexer using (a) ordinary gates, (b) gates with 3-state outputs, and (c) transmission gates. Under what circumstances would (c) be preferable?

You might wonder what to do if you want to select among more inputs than are provided in a multiplexer. This question comes under the general category of chip "expansion" (using several chips that have small individual capabilities to generate a larger capability), and it applies to decoders, memories, shift registers, arithmetic logic, and many other functions as well. In this case the job is easy (Fig. 8.34). Here we have expanded two 74LS151 8-input multiplexers into a 16-input multiplexer. There's an additional address bit, of course, and you use it to enable one chip or the other. The disabled chip holds its Q LOW, so an OR gate at the output completes the expansion. With three-state outputs the job is even simpler, since you can connect the outputs directly together.

Demultiplexers and decoders

A demultiplexer takes an input and routes it to one of several possible outputs, according to an input binary address. The other outputs are either held in the inactive state or open-circuited, depending on the type of demultiplexer.

A decoder is similar, except that the address is the only input, and it is "decoded" to assert one of n possible outputs. Figure 8.35 shows an example. This is the '138 "1-of-8 decoder." The output corresponding to (addressed by) the 3-bit input data is LOW; all others are HIGH. This particular decoder has three ENABLE inputs, all of which must be asserted (two LOW, one HIGH); otherwise all outputs are HIGH. A favorite use of the decoder is to cause different things to happen, depending on the state of a "counter" chip that drives it (more on this, soon). Decoders are commonly used when interfacing to microprocessor, to trigger different actions depending on the address; we'll treat this subject in detail in Chapter 10. Another common use of a decoder is to enable a sequence of actions in turn, according to an advancing address given by the output of a binary *counter* (Section 8.25). A close cousin of the '138 is the '139, a dual 1-of-4 decoder with a single LOW-true enable per section. Figure 8.36 shows how to use a pair of '138 1-of-8 decoders to generate a 1-of-16 decoder. No external gates are necessary,

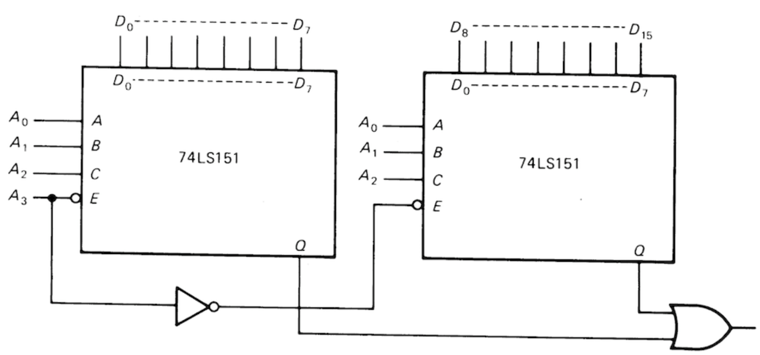


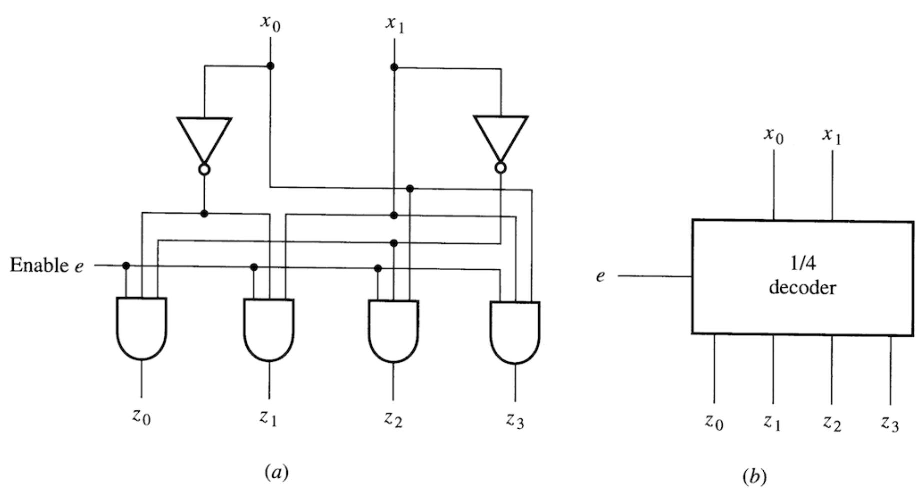
Figure 8.34. Multiplexer expansion.

(*See* Horowitz and Hill, The Art of Electronics, Second Edition, 9, page 496, produced asGPH_DE_01696551 – GPH_DE_01696655, attached herein as Exhibit F.)

Decoders. A 1-out-of-2^n or $1/2^n$ *decoder* is a combinational circuit with n input lines X and 2^n output lines Z such that each of the 2^n possible input combinations A_i applied to X activates a corresponding output line z_i. Figure 2.23 shows a 1/4 decoder. Several $1/2^n$ decoders can be used to decode more than n lines by connecting them in a tree configuration analogous to the multiplexer tree of Figure 2.21. The main application of decoders is address decoding, where A_i is interpreted as an address that selects a specific output line z_i or some circuit attached to z_i. For example, decoders are used in RAMs to select storage cells to be read from or written into.

Another common application of decoders is that of routing data from a common source to one of several destinations. A circuit of this kind is called a *demultiplexer,* since it is, in effect, the inverse of a multiplexer. In this application the control input e (enable) of the decoder is viewed as a 1-bit data source to be routed to one of 2^n destinations, as determined by the address applied to the decoder. Thus a $1/2^n$ decoder is also a 2^n-output, 1-bit demultiplexer. A k-output, m-bit demultiplexer can be readily constructed from a network of decoders. Figure 2.24 shows a four-output, 2-bit demultiplexer that employs two 1/4 decoders of the type in Figure 2.23.

Encoders. An *encoder* is a circuit intended to generate the address or index of an active input line; it is therefore the inverse of a decoder. Most encoders have 2^k input data lines and k output data lines. For example, when $k = 3$, entering a data



(*See* John P. Hayes, Computer Architecture and Organization, Third Edition, 1998, page 90, produced asGPH_DE_01696656 – GPH_DE_01696710, attached herein as Exhibit G.)

45. The decoder is almost exactly the inverse of a selector circuit, which the court considered in S3 v. Nvidia. In my opinion the selector and decoder have equal complexity, both are taught to undergraduates and require roughly the same degree of hardware, both are

represented schematically using a quadrilateral symbol, and both were well known to a person of ordinary skill in the art.

46. All parties have agreed that the claimed instruction decode means function should be construed as "[r]esponsive to each two parcel item, determining in one clock cycle whether the two-parcel item is a single two length of 2n bits, or two one-parcel instructions, each having a bit length of n bits." As I just described, a person of ordinary skill in the art would know that a digital logic decoder circuit, of the sort described above, was the corresponding structure. Furthermore, the '881 Patent specification indicates that "[t]he present improvement decodes all 64 bits at once, examines the instruction fields, determines whether the 64 bits are one or two instructions, determines which instruction goes where and whether it goes on the first or second issue cycle." (*See* GPH Brief at Exhibit 1, '881 Patent at 27:55-59.) This passage thus indicates that the structure is two decoders, of the sort discussed in the two textbook passage appearing above, connected to the two opcode fields in the two parcels.

47. Moreover, while it is unnecessary to those skilled in the art, the '881 Patent describes an embodiment of the internal structure of the hardwired instruction decoder 122a in the detailed instruction set architecture that is provided in Appendices A-B of the '881 Patent. Those skilled in the art would understand that the internal design of hardwired logic, such as that of the hardwired instruction decoder 122a, are typically expressed using one or more of truth tables, Boolean equations, or instruction sets.

48. By way of example, a truth table discloses the internal structure of a digital circuit by expressing the relationship between the inputs and outputs of a digital logic circuit in tabularized form that is used by digital logic designers. Consider the truth table shown below. The first column shows values of four binary inputs including all possible sixteen

combinations in numerical order. When there are exactly sixteen possible combinations the values are considered to be in hexadecimal format and are also represented as 0x0 through 0xF. The second through seventeenth columns each hold one output value labeled D0 through DF. Each of these output values produces a value of 1 for one and only one input value. I have marked the each input value in red in the chart below. This truth table corresponds to a 4-bit decoder.

Value	**D0**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**	**D9**	**D A**	**DB**	**D C**	**D D**	**DE**	**DF**
0000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0010	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0011	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0100	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0101	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0110	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0111	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1000	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1001	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1010	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1011	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1100	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1101	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

49. However, as I mentioned earlier, there are many other ways to express the same information that discloses the internal structure of a digital logic decoder, albeit in a different form of representation. One such way is by providing a detailed description of the instruction set architecture. Those skilled in the art would understand that, without any additional information, such a detailed instruction set architecture is equivalent to a truth table, logic diagram, Boolean equation, or any other form of expression that they so desire.

50. Defendants ignore the fact that the ‘881 Patent provides the information that is needed to disclose internal structure of the hardwired instruction decoder 122a. In particular, the ‘881 Patent provides a detailed instruction set architecture, which provides the same information that would be contained in a truth table, but expressed in a different form. This information provides a complete description of the internal structure of the digital logic decoder circuit that carries out the function of the claimed instruction decode means.

51. In particular, Appendix B of the ‘881 Patent specifies the instruction set of the preferred embodiment. Consider one example page such as bb-15. (*See* Exhibit E, Excerpts from ‘881 Patent Microfiche, produced as GPH_DE_01696219 - GPH_DE_01696550 at GPH_DE_01696476.) The top of the page indicates that this is the LOAD instruction and the top right indicates that the encoding must follow the form “bb xx jj qq.” This encoding specifies, among other things, that the first four bits of the instruction have the value of “b” or 0xB in hexadecimal or 0x1011 in binary. Furthermore it specifies that the instruction is one parcel in length. Now consider page Bc-1, which describes the JLI instruction. Near the top of the page the encoding for this instruction is shown to be “c0 tt jj kk nnnnnnnn.” (*See Id.* at GPH_DE_01696478.) This encoding specifies, among other things, that the first four bits of the instruction have the value of “c” or 0xC in hexadecimal or 0x1100 in binary. The encoding specifies that the instruction is two parcels in length.

52. Appendix B of the ‘881 Patent indicates that each two-parcel instruction begins with the values "c’, ‘d’, ‘e’, or ‘f.’ Thus the structure will determine a two-parcel instruction is present if the first four bits of a two-parcel item are ‘c’ OR ‘d’ OR ‘e’ OR ‘f’. This logic function is necessary, unique, and fully constrained by the ‘881 Patent specification.

Furthermore, using the logic table for the 4-bit decoder above, a person of ordinary skill in the art would immediately know that the specified digital logic structure is OR (DC, DD, DE, DF).

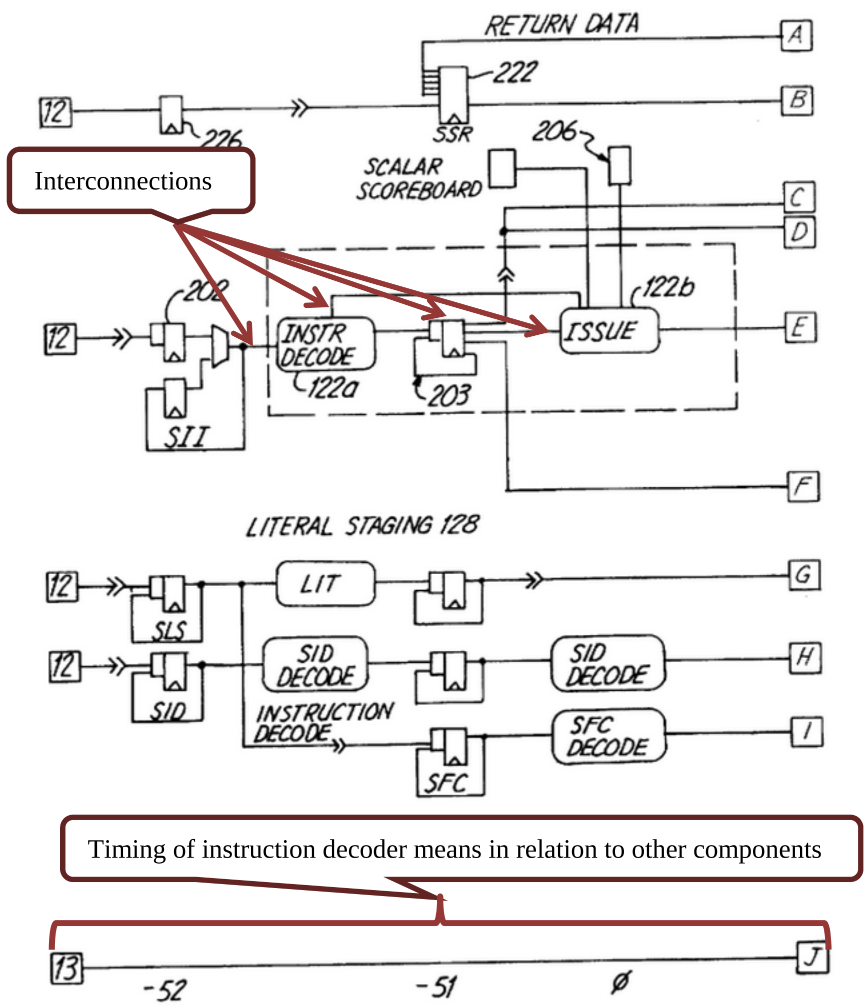
53. The ‘881 Patent indicates that when the two-parcel item does not contain a two-parcel instruction there will be a one parcel instruction in the first parcel (*See* ‘881 Patent at Fig. 4A and fig. 4B). The uniquely and fully specified logic structure for identifying this situation is NOT(a two-parcel instruction) or NOT(OR(DC,DD,DE,DF)). Furthermore, when the first parcel is a one-parcel instruction and the second parcel is a no-op (*see* GPH Brief at Exhibit 1, at Fig. 4B) the first four bits of the second parcel will be ‘0’ (*see* Exhibit E, Excerpts from ‘881 Patent Microfiche at GPH_DE_01696254). Thus, the case where there are two one-parcel instructions corresponds uniquely to the specified internal logic structure of AND(NOT(OR(DC,DD,DE,DF)),NOT(D0’)) where D0’ is the D0 output for a 4-bit decoder, of the same sort known to one of ordinary skill in the art and specified structurally by the ‘881 Patent, that is connected to the first 4 bits of the second parcel in the instruction register.

54. Thus, in my opinion, the ‘881 specification clearly and unambiguously specifies a decode structure corresponding to two 4-bit decoders, one OR-gate, one AND-gate, and two NOT-gates producing the two functions claimed in the decode means element with the structure discussed above. All of these components were well known to a person of ordinary skill in the art and their selection and organization is specified and fully constrained by the ‘881 specification. While the logic circuits that I have laid are directly structurally derived by the constraints of the ‘881 specification, any equivalent alternate organization would necessarily satisfy the same truth-table.

55. The appendix provides a complete definition of the instruction encoding and semantics, not simply “some indication.” Each instruction is associated with one, and only

one, opcode. Each opcode is specifically identified as one or two parcels in length. As I have already stated, while it is true that the appendix does not draw out the logic diagram used for decoding this is because a person of ordinary skill in the art would understand that the logic diagram is that of the digital logic decoder circuit 122a shown in Figure 7C. (*See* '881 Patent at Figure 7C.)

56. Moreover, the specification shows numerous other details regarding the periphery to the structure of the hardwired instruction decoder 122a. For example, the specification shows that the opcode fields are uniquely associated with instruction length, defines the bits that form the opcode, and the specification states that the instruction fields feed the decoder. Moreover, how the digital logic decoder circuit communicates between elements and the timing is specified in Figure 7C (reproduced below).



(*See* GPH Brief at Exhibit 1, '881 Patent at Fig. 7C.)

57. Thus, in my opinion, the '881 Patent discloses a sufficiently definite structure corresponding to the instruction decode means function.

D. One Clock Cycle

Claim Language	GPH's Construction	Defendants' Construction
"One clock cycle"	A measurement of a single unit of synchronizing time of a processor	A single instruction pipeline clock cycle defined by the period between successive rising edges of the clock signal.

58. GPH suggests that the term "one clock cycle" should be construed to mean "a measurement of a single unit of synchronization time of a processor ." Defendants propose to construe this term to mean "a single instruction pipeline clock cycle defined by the period between successive rising edges of the clock signal." In my opinion GPH has proposed the

proper construction. The claim term is precise in that it defines a number of clock cycles, one, as opposed to a particular or specific clock cycle out of the multiplicity of clock cycles over time. Computer systems typically employ a number of clocks operating at different frequencies. Such an approach is often even employed within the instruction pipeline and was well known at the time the '881 Patent was filed. It is not surprising that the claim language speaks simply in terms of the number of clock cycles without overly constraining the claim by identifying other properties of that clock signal (e.g. *Id.,* '881 Patent at 2:65-3:17). Thus I believe GPH has accurately represented the limitations envisioned by the patentee.

59. In contrast, Defendants have introduced the notion of an "instruction pipeline clock cycle" into their construction. The phrase "instruction pipeline clock" does not appear within the specification of the '881 Patent, thus Defendants' construction has no support within the specification. That this phrase does not appear in the patent does not surprise me since, as I mentioned, the patentees were certainly familiar with the approach of utilizing multiple clocks of varying frequencies. Defendants' construction would inappropriately limit the claim to instruction pipelines that only utilize a single clock frequency. It would also restrict the claim term to referring to a particular clock cycle out of the many clock cycles over time, as opposed to the duration of a clock cycle. Interpreting the term "one clock cycle" in that manner is counter to the plain meaning of the term. "One clock cycle" describes a period of time; it does not identify a particular or specific clock cycle. I have not been presented with any support for this position but would prefer to reserve my right to respond if such evidence is made available to me in the future. Moreover, Defendants' proposed construction that the single instruction pipeline clock cycle should be defined between successive rising edges of the clock signal is not supported by the specification, confusing, and not helpful. As a preliminary matter, neither the

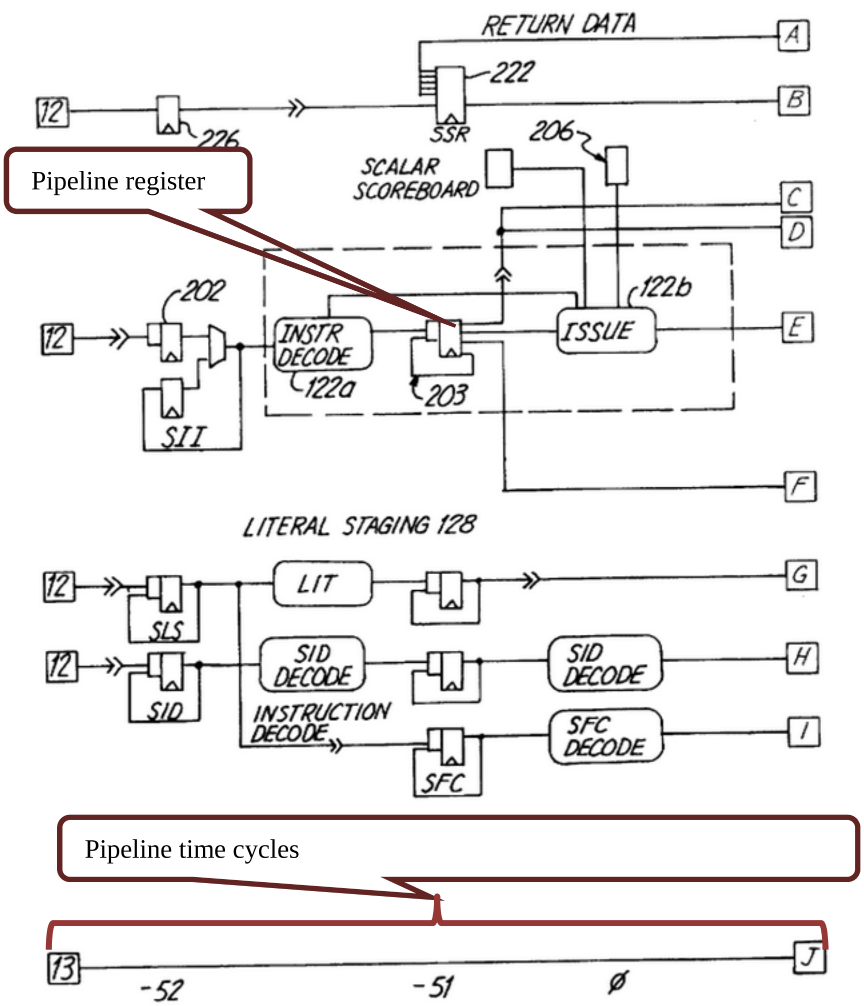
specification nor the claims make any mention of the clock cycle being defined by rising or falling edges of a clock signal. Indeed, those of ordinary skill would understand that whether a clock cycle is defined by rising edges or falling edges is a mere convention of no significance. For example, those skilled in the art would understand that a clock cycle could be defined by successive rising edges, successive falling edges, or according to a number of other conventions. Moreover, Defendants' construction is confusing because it is not clear whether "the clock signal" is referring to the claimed clock cycle, a pipeline time cycle, Defendants' "instruction pipeline clock cycle," or something else altogether.

60. Moreover, the claim does not state that the same instruction is decoded and issued in the same clock cycle. Rather, the claim states that during "one clock cycle" a two-parcel instruction or two one-parcel instructions are decoded and a two-parcel instruction or a first one-parcel instruction is issued. If the patentee had intended for the same instruction to be decoded and issued during a single clock cycle they would most likely have drafted claim one to use the phrase "issuing said two parcel instruction," referring back to the same instruction discussed in the decoding element, rather than claiming "issuing each two parcel instruction." Similarly the phrase "issuing one then the other of the two one parcel instructions" would have been drafted to say "issuing one then the other of the said two one parcel instructions." Defendants have incorporated a limitation into claim 1 that is not present in the claim or supported by the intrinsic record.

61. Furthermore, the '881 Patent specification indicates that the interpretation suggested by Defendants cannot reflect the inventors' clear intent. The specification states: "An instruction execution unit 120 in the scalar means 102 includes decode and issue means 122, branch logic means 124, a program counter (PC) register 126 and literal transfer means 128."

(*See Id.* '881 Patent, at 4:59-62). Element 122 appears in Fig. 7c-7d. These figures show pipeline stages separating the instruction decode means 122a and the instruction issue unit 122b. Pipelining is a common technique for constructing computer circuits and involves a combination of logic circuits and pipeline registers. The logic circuits are connected together to calculate a desired function, such as the addition of two numbers. The pipeline registers are used to hold the results (or data) of the logic circuits stable so that the results may be reliably used as inputs for subsequent logic circuits. This technique is often illustrated as a linear array of logic circuits and pipeline registers with inputs arriving from the left and outputs being produced to the right. In such an illustration a vertical segment of the figure will correspond to the logic circuits that are simultaneously operating on a block of data. Such a group of logic circuits is referred to as a pipeline stage may be labeled with the variable "s" with the preceding stage labeled "s-1" and the subsequent stage labeled "s+1". Similar though different approaches are known to be used to indicate the relative timing relationship between stages in a pipeline.

62. Defendants' proposal would exclude the preferred embodiment of the '881 Patent (shown below), which places a pipeline register between the decode means and the issue means, such that they do not decode and issue an individual instruction in a "instruction pipeline clock cycle." (*See Id.* , at Fig. 7C.)



(*See Id.* , at Fig. 7C.)

E. Instruction Issue Means Responsive To The Instruction Decode Means For Issuing Each Two Parcel Instruction For Execution During Said One Clock Cycle, And For Issuing One Then The Other Of The Two One Parcel Instructions For Execution In Sequence During Said One Clock Cycle And The Next Succeeding Clock

Claim Language	GPH's Construction	Respondent's Construction
instruction issue means responsive to the instruction decode means for issuing each two parcel instruction for execution during said one clock cycle, and for issuing one then the other of the two one parcel instructions for execution in sequence during said one clock cycle and the next succeeding clock	This phrase should be construed pursuant to 35 U.S.C. § 112, ¶ 6. Function: Responsive to the instruction decode means, issuing each two parcel instruction during one clock cycle and issuing one then the other of the two one parcel instructions in sequence during said one cycle and the next succeeding clock cycle.	This term should be construed in accordance with 35 U.S.C. § 112, ¶6. Recited Function: Responsive to the instruction decode means, issuing each two parcel instruction during one clock cycle and for issuing one then the other of the two one parcel instructions in sequence during said one cycle and the next succeeding clock cycle.

Claim Language	GPH's Construction	Respondent's Construction
	Structure: Hardwired instruction issue mechanism 122b and holding means 203 an embodiment of which is further described by 3:4-12, 4:59-62, 9:41-10:1, 27:47 - 52, 33:1-5, 33:44-46, and Figs. 2 and 7C.	Corresponding Structure: The specification does not disclose a corresponding structure, therefore this limitation is indefinite under 35 U.S.C. § 112.

63. GPH has suggested that the function for this claim term is essentially the function as recited within the term. Defendants' appear to have rearranged the language a bit, but have essentially proposed an equivalent function. It is my opinion that both identified functions are satisfactory for claim construction from the point of view of a person of ordinary skill in the art. GPH has proposed that the "instruction issue means" term be construed with the hardwired instruction issue mechanism 112b and holding means 203 serving as the corresponding structure.

64. As a preliminary matter there can be no serious dispute as to whether instruction issue mechanism 122b and holding means 203 is the structure corresponding to the claimed issue means function. The '881 Patent specification unambiguously provides that "[t]he instruction cache 110 delivers instructions to four places. One is the instruction <u>decode and issue means 122a and 122b including the holding means before 122a, with the actual instruction issue decision making logic means being 122b</u>." (*See Id.* at 33:1-5 (emphasis added)).

65. By way of further example, the '881 Patent specification indicates that element 122b is an "issue portion" of an "issue mechanism": "The issue mechanism includes a decode portion 122a and an issue portion 122b. A single stage holding means 203 couples the decode and issue mechanisms." (*Id.* at 9:41-45). Moreover, in the context of a discussion of trap instructions, the specification states "Simultaneously, the trap instruction goes into the issue mechanism through the decode means 122a and issue means 122b." (*Id.* at 33:44-46). In my opinion the claim has clearly identified a function, the specification has clearly stated at least one

structure for implementing that function, and a person of ordinary skill in the art would have no difficulty understanding the function claimed and corresponding structure disclosed.

66. Defendants' position that the '881 Patent does not disclose structure corresponding to the instruction issue means ignores that the '881 Patent clearly constrains the structure for the instruction issue means to be a hardwired issue circuit. See Exhibit D, Excerpts of '881 Patent File History, March 19, 1997 Amendment and Response to Office Action & Examiner Interview Summary, produced as GPH_DE_01511506 – GPH_DE_01511512 , at 6 ("When the control functions of a digital computer system are generated by hardware using hardware logic design techniques, the control unit is said to be 'hardwired.") For the same reasons previously discussed with respect to the instruction decode means, the structure corresponding to the instruction issue means 122b is a hardwired issue circuit. As was the case with the instruction decode means, it is understood by one of ordinary skill in the art that the instruction issue means must be hardwired, i.e. implemented with digital logic circuits.

67. Moreover, the '881 goes so far as to disclose the internal structure of the digital logic circuit corresponding to the instruction issue means. For example, '881 Patent specification discloses that "[i]f the issue mechanism determines that the operand data and L registers required by the instruction are in fact available for use, the issue mechanism 122b provides an issue output on line 122c." (*See* GPH Brief at Exhibit 1, '881 Patent at 9:55-58.)

68. As previously discussed, hardware design is often described using logical and propositional logic statements in code or short hand. To a person of ordinary skill in the art an "if" statement, as disclosed above, is equivalent to an AND-gate where one input to the AND gate is the signal that may or may not pass through (i.e. the decoded instruction) and the other is the controlling signal (the operand data via the disclosed register scoreboard.) (*See Id.* at 9:49-

58.) Thus, this passage clearly indicates that the corresponding structure is an AND gate that either blocks the propagation of the decoded instruction until a future time or issues it to the execution units.

69. In the alternative, the Defendants have also suggested that if the term is not indefinite the corresponding structure should be identified as the issue mechanism 122b, counter 206 and queue 132. I believe they have identified the wrong structures and will explain my reasoning.

70. The following passage contains all references to counter 206 that exist within the specification of the ‘881:

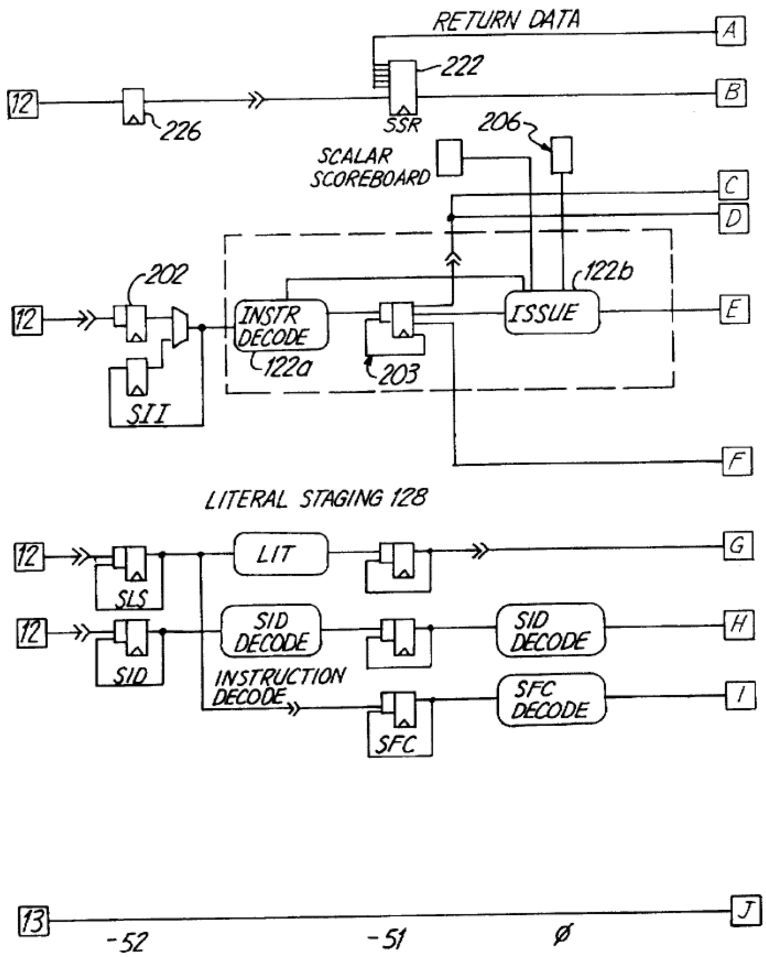
> *"In the event that the issue mechanism 122B determines from the decoded instruction that it is a vector instruction, it determines from a counter 206 whether or not a buffer queue 132 for storing vector instructions is full. If the instruction decoded is a scalar/vector instruction, i.e., it calls for a scalar register as a source of an operand or a memory address, then the issue mechanism determines from the scalar register scoreboard whether the required scalar data is available and also determines from the counter 206 whether or not the queue 132 is able to accept another vector or vector/scalar instruction. If the scalar data is available and the queue 132 can accept the instruction, the scalar/vector instruction is transferred to the queue 132."*

(See Id. at 9:61-65.)

71. This passage clearly identifies that counter 206 and queue 132 are not part of the issue mechanism. Furthermore 206 and 132 serve roles within the execution of vector instructions, not scalar instructions. As I mentioned above it is not possible to achieve a sustained rate of issuing one instruction per clock cycle within the vector processor.

72. I have reproduced Fig. 7c from the ‘881 Patent below in order to revisit observations made previously in the context of the Instruction Decode Means. This figure clearly discloses that the issue mechanism 122b is fed by the scalar scoreboard which has a registered output (*Id.* at 9:52-55), the counter 206 which has a registered output, and the

instruction decode mechanism which has a registered output. For each of these inputs to affect the issue mechanism 122b at clock cycle 0 they must have been updated no later than clock cycle –S1.



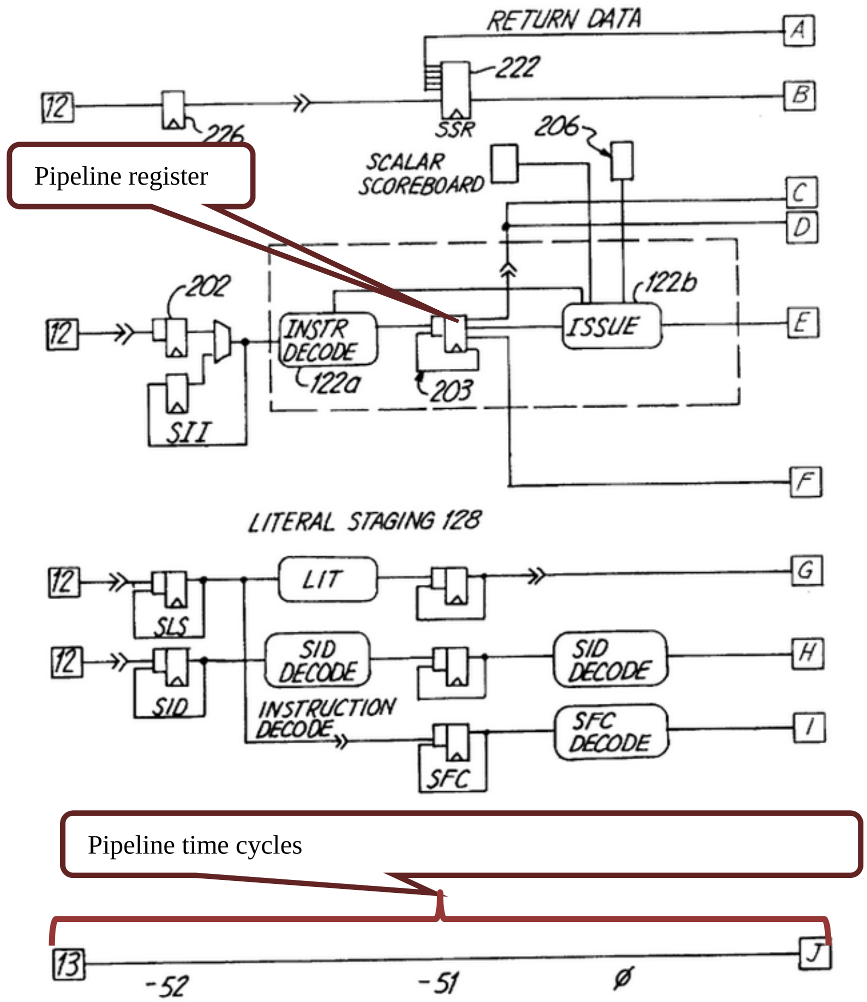
73. Defendants even cite a passage that undercuts their proposed construction: “If the queue is not full, the vector instruction is issued and transferred to the queue 132.” (*Id.* at 9:67-10:1). Clearly when the vector instruction arrives at queue 132 it has been issued, thus 132 is not part of the means for issuing instructions.

F. Said One Clock Cycle

Claim Language	GPH’s Construction	Respondent’s Construction
“Said one clock cycle”	See “one clock cycle”	The same single instruction pipeline clock cycle

74. GPH suggests referring to its proposed constructions for the term “one clock cycle” which was addressed previously. I agree with its position. The term “said one clock cycle” does not require additional construction beyond what is necessary for determining the meaning of the phrase “one clock cycle.” This term refers to a duration of time, i.e. the duration of one clock cycle, not an instant of time, i.e. a particular clock cycle, as urged by Defendants.

75. Defendants have proposed to construe this term to mean “the same single instruction pipeline clock cycle.” As I have already discussed, this construction is flawed because it relies upon the flawed foundation of Defendants’ proposed construction for “one clock cycle,” which relies upon the ill-defined and non-intrinsic term “instruction pipeline clock cycle.” However, ignoring this liability, I am not certain what the import of Defendants’ proposed construction is. They say the term should mean “the same single instruction pipeline clock cycle” but they do not identify which specific clock cycle they are referring to. This is a limitation without narrowing effect and thus I am at a loss to see what value it would provide to the court for interpreting these terms. To the extent that the Defendants are referring to the pipeline time cycles shown in Fig. 7C (reproduced below), then Defendants’ proposed construction would purport to exclude a preferred embodiment of the instruction issue means, which places a one pipeline time cycle delay in between the decode and the issuance of every single instruction.



(*See Id.* at Fig. 7C.)

G. Hardwired

Claim Language	GPH's Construction	Respondent's Construction
"Hardwired"	The control functions for a digital computer system are generated by hardware using hardware logic design techniques.	The functions for a digital computer system are generated by hardware using hardware logic design techniques. ACI: A data processing apparatus that carries out operations directly in response to instructions without decomposing any operations into sub-operations.

76. All of the parties except for ACI appear to agree that the term hardwired should be construed according a phrase occurring in multiple places within the file history of the ‘881 Patent: “When the control functions of a digital computer system are generated by hardware using hardware logic design techniques, the control unit is said to be ‘hardwired.’” All of the proposed constructions appear to be relatively close to the language employed by the inventors with a few differences. Indeed, those skilled in the art would understand that the control functions of the instruction decode and issue means are generated by hardware using hardware logic design techniques.

77. ACI’s proposed construction for “hardwired” contradicts both the intrinsic record of the ‘881 Patent and the way in which the term would be understood by those skilled in the art. By way of example, the ‘881 Patent sets forth various instructions in Appendix B that cause the processor to decompose operations into suboperations. One example is the “NOR” instruction, which the processor decomposes into the suboperations of “[p]erform[ing] the logical nor of (sk) with each element of vj, *and* mov[ing] the result into the corresponding element of vi.” See *Id.*, Appendix B at B7-32 (emphasis added.)

78. Consequently, it is my opinion that the word “hardwired” should be construed to mean “The control functions for a digital computer system are generated by hardware using hardware logic design techniques” as proposed by GPH.

VI. CONCLUSION

79. Having reviewed the ‘881 Patent, Prosecution History, and the Joint Submission of Proposed Claim constructions, I have reached certain opinions concerning the proper constructions for contested terms of the ‘881 Patent.

80. It is my opinion that GPH and Defendants disagree on many of the terms under consideration. After careful consideration it is my opinion that a person of ordinary skill

in the art would find the terms discussed both definite and accurately construed using the proposals suggested by GPH.

81. I am fully prepared to provide testimony under oath with regards to the issues raised in this declaration.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct to the best of my knowledge.

Executed on: February 7, 2014

Dr. William Henry Mangione-Smith